

Manipulasi data dalam SQL mencakup berbagai perintah untuk menambahkan, mengubah, menghapus, dan membaca data dari tabel database. Ada beberapa poin yang perlu diperhatikan dalam manipulasi data SQL, yaitu INSERT, UPDATE, DELETE, SELECT, WHERE, ORDER BY, dan LIMIT.



1. Menambahkan Data (INSERT)

Perintah INSERT digunakan untuk menambahkan data baru ke dalam tabel. Data yang dimasukkan harus sesuai dengan struktur tabel yang telah dibuat.

- Sintaks dasar:

```
INSERT INTO nama_tabel (kolom1, kolom2, ...) VALUES (nilai1, nilai2, ...);
```

- Jika semua kolom dalam tabel akan diisi, kita bisa menghilangkan bagian daftar kolom, seperti berikut:

```
INSERT INTO siswa (id, nama, usia) VALUES (1, 'Budi', 17);
```

Perintah ini menambahkan siswa dengan ID 1, nama "Budi", dan usia 17 tahun ke dalam tabel.

Catatan: Pastikan nilai yang dimasukkan sesuai dengan tipe data kolom dalam tabel. Jika ada kolom dengan tipe data INTEGER, maka nilai yang dimasukkan juga harus berupa angka.

2. Mengubah Data (UPDATE)

Perintah UPDATE digunakan untuk mengubah data yang sudah ada dalam tabel berdasarkan kondisi tertentu.



- Sintaks dasar:

`UPDATE nama_tabel SET kolom1 = nilai_baru WHERE kondisi;`

Jika WHERE tidak digunakan, maka semua data dalam tabel akan diperbarui.

- Contoh:

- Misalkan kita ingin mengubah usia siswa dengan ID 1 menjadi 18 tahun:

`UPDATE siswa SET usia = 18 WHERE id = 1;`

Catatan: Selalu gunakan WHERE agar hanya data yang diinginkan yang diperbarui. Jika tidak, semua baris dalam tabel akan berubah.

3. Menghapus Data (DELETE)

DELETE digunakan untuk menghapus data dari tabel.

- Sintaks dasar:

`DELETE FROM nama_tabel WHERE kondisi;`

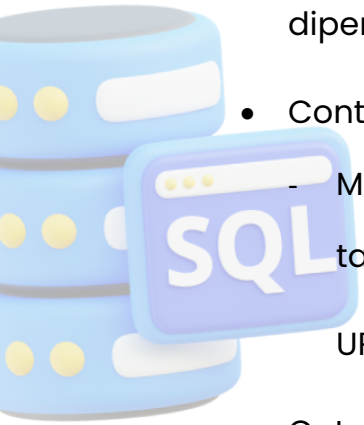
Jika WHERE tidak digunakan, semua data dalam tabel akan dihapus.

- Contoh menghapus siswa dengan ID 1:

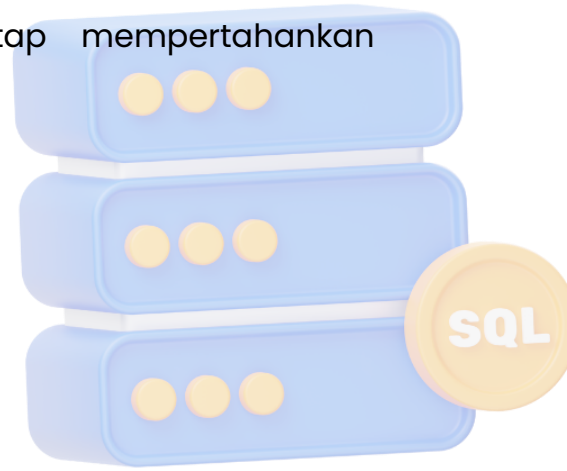
`DELETE FROM siswa WHERE id = 1;`

Catatan: Hati-hati dalam menggunakan DELETE. Jika ingin menghapus seluruh isi tabel tetapi tetap mempertahankan strukturnya, gunakan:

`DELETE FROM nama_tabel;`



SQL&Elearn



- Jika ingin menghapus seluruh data dan mereset auto-increment, gunakan:

```
TRUNCATE TABLE nama_tabel;
```

4. Membaca Data (SELECT)

Perintah SELECT digunakan untuk mengambil dan menampilkan data dari tabel.

- Sintaks dasar:

```
SELECT kolom1, kolom2 FROM nama_tabel;
```

Untuk menampilkan semua kolom, gunakan *:

```
SELECT * FROM nama_tabel;
```

- Contoh:

Menampilkan nama dan usia dari tabel siswa:

```
SELECT nama, usia FROM siswa;
```

- Menampilkan seluruh data siswa:

```
SELECT * FROM siswa;
```



5. Filtering Data dengan WHERE

WHERE digunakan untuk mengambil data yang memenuhi kondisi tertentu.

- Sintaks dasar:

```
SELECT FROM nama_tabel WHERE kondisi;
```

- Contoh:

- Menampilkan siswa yang berusia lebih dari 17 tahun:

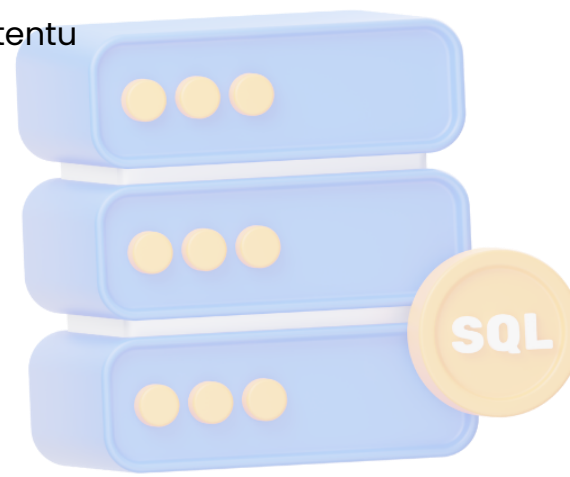
```
SELECT FROM siswa WHERE usia > 17;
```

- Menampilkan siswa dengan nama "Budi":

```
SELECT * FROM siswa WHERE nama = 'Budi';
```

Operator yang sering digunakan dalam WHERE:

1. = : Sama dengan
2. != atau <> : Tidak sama dengan
3. > : Lebih besar
4. < : Lebih kecil
5. >= : Lebih besar atau sama dengan
6. <= : Lebih kecil atau sama dengan
7. LIKE : Mencari pola tertentu (misalnya nama yang mengandung kata tertentu)
8. IN : Memilih beberapa nilai sekaligus
9. BETWEEN : Memilih data dalam rentang tertentu



- Contoh penggunaan LIKE:

```
SELECT * FROM siswa WHERE nama LIKE 'B%';
```

Perintah ini menampilkan semua siswa yang namanya dimulai dengan huruf "B".



6. Pengurutan Data dengan ORDER BY

Klausul ORDER BY digunakan untuk mengurutkan data berdasarkan kolom tertentu dalam urutan naik (ASC) atau turun (DESC).

- Sintaks dasar:

```
SELECT FROM nama_tabel ORDER BY kolom ASC|DESC;
```

- Contoh:

- Mengurutkan siswa berdasarkan nama secara alfabetis:

```
SELECT FROM siswa ORDER BY nama ASC;
```

- Mengurutkan siswa berdasarkan usia dari yang tertua ke termuda:

```
SELECT * FROM siswa ORDER BY usia DESC;
```

Catatan: Jika tidak menentukan ASC atau DESC, maka defaultnya adalah ASC (naik).

7. Membatasi Data dengan LIMIT

LIMIT digunakan untuk membatasi jumlah data yang ditampilkan.

- Sintaks dasar:

```
SELECT FROM nama_tabel LIMIT jumlah_data;
```



- Contoh:

Menampilkan 5 siswa pertama dalam tabel:

```
SELECT FROM siswa LIMIT 5;
```

- Menampilkan 3 siswa tertua:

```
SELECT * FROM siswa ORDER BY usia DESC LIMIT 3;
```

Catatan: LIMIT sangat berguna saat mengambil data dalam jumlah besar agar lebih efisien.

SQL&Earn

